Cassandra Crossing/ La fine del mondo, di silicio

(577)—Cosa c'è nelle CPU ed in tutti i chip che fanno girare il mondo? Solo quello che c'è scritto nei datasheet? No, c'è molto di più…

Cassandra Crossing/ La fine del mondo, di silicio

Figure 1: Dr. Strangelove trailer from 40th Anniversary Special Edition DVD, 2004. Quest'opera è nel pubblico dominio perché pubblicata negli Stati Uniti fra il 1929 e il 1977, inclusi, senza un avviso di copyright. https://commons.wikimedia.org/wiki/File:Dr._Strangelove_-_The_War_Room.png

(577)—Cosa c'è nelle CPU ed in tutti gli altri chip che fanno girare il mondo? Solo quello che c'è scritto nei datasheet? No, c'è molto di più, e la complessità è, come sempre, pericolosa.

8 marzo 2024—Cosa succederebbe se l'intera informatica di tutto il mondo si fermasse? Se i computer, tutti assieme, si bloccassero, o peggio si mettessero a fare altro?

I lettori di Cassandra, reduci dalla lettura della prima parte di questa miniserie, potranno cogliere il collegamento, mentre gli altri sono vivamente invitati a fermarsi un attimo e leggerla.

Talvolta ce lo dimentichiamo; al mondo non è il software che fa succedere le cose, ma sono quei pezzetti di silicio finemente inciso e stampato, altrimenti noti come "circuiti integrati", o per quelli incapaci di parlare di tecnologia in italiano, "chip".

I più importanti sono le CPU, dall'Intel 4004 in poi. E siamo abituati a pensare che le CPU siano oggetti monolitici, che funzionano in un determinato modo e non possono assolutamente essere sovvertiti.

“*Madornale errore*”—direbbe il nostro pluricitato amico Jack Slater. E, probabilmente, mai come in questo caso avrebbe ragione.

La causa prima di tutto questo è lo spreco di transistor, reso possibile dalle tecnologie di produzione dei circuiti integrati. Si riescono a fare sempre più piccoli, sempre più economici, e i circuiti integrati moderni ne hanno un numero sempre maggiore.

I numeri delle CPU fanno spavento. Nel 1976 uno Z-80 si accontentava di 8.500 transistor, mentre oggi una CPU Apple M2 Ultra ne ha 67.000.000.000, ed Intel ha in produzione Wafer Scale Engine 2, un’architettura di chip che permette di realizzare CPU fino a 2.600.000.000.000 transistor. Si, parliamo di *trilioni* di transistor!

Non penserete mica che servano ad implementare una cosa semplice come una CPU “pura”?

No, sono decenni che le CPU commerciali sono in realtà macchine molto, molto, ma davvero molto più complesse, che si comportano “normalmente” come CPU in virtù di una architettura sottostante molto più elaborata, basata su microcodici, che è in parte “programmabile” per modificarne le funzionalità.

Facciamo un esempio, ormai datato (2005) ma ben esemplificativo del problema. Tutte le CPU Intel prodotte negli ultimi 15 anni contengono ME, un Management Engine (motore di gestione) che permette di “amministrare” un pc e *fare cose* anche quando il pc è spento, anche senza hard disk, anche quando è guasto, purché abbia l’alimentazione elettrica. Ed ovviamente anche mentre è acceso, anche mentre una persona ci sta lavorando.

Rileggete la frase precedente, e tremate.

E questa è una funzionalità “pubblica”, pubblicizzata e venduta come funzionalità “amministrativa”; ed in effetti, in certi ambiti aziendali, può davvero essere utile.

“*Sic stantibus rebus*” Cassandra non osa nemmeno immaginare cosa altro sia senz’altro presente nelle CPU dei nostri pc, senza che nemmeno la maggior parte degli esperti informatici lo sappia.

Ma torniamo a noi ed alla funzionalità *Intel Management Engine*. Per spiegare come sia possibile realizzarla è necessario avere accesso a documentazione semipubblica, riuscire a capirla ed a riassumerla.

Chi volesse affrontare una lettura un po’ tecnica della questione, potrebbe leggersi questo articolo, da tempo scomparso dal web ma saldamente memorizzato su quella inestimabile risorsa che è Internet Archive.

Per tutti gli altri ed in parole semplici; le CPU Intel moderne girano i programmi in una struttura di gerarchie di esecuzione—dette Ring - nella quale i programmi *normali* girano a Ring 3.

Tutto quello che gira ad un livello inferiore ha il completo controllo di quello che gira ad un livello superiore. Così alcune parti delle applicazioni e del sistema operativo girano a Ring 2, la maggior parte del sistema operativo gira a Ring 1, ed a Ring 0 troviamo i programmi che girano *davvero* sulla CPU, come il *kernel* ed il gestore di memoria virtuale.

Possiamo banalizzare dicendo che il Ring 0 è la “vera” CPU, e che solo i programmi che girano a Ring 0 hanno il completo accesso alla CPU stessa.

Ma questa “CPU”, a sua volta, è un oggetto parzialmente programmabile, che in realtà gira “microcodici”, i quali possono essere modificati ed aggiornati. E questo avviene a livelli di Ring **negativi**, *sotterranei*.

Ed a **Ring -3**, ben nascosto da occhi plebei, troviamo MINIX, un intero sistema operativo, memorizzato nel silicio, che gira allegramente ben oltre ogni nostra possibilità di esame, e che, tra l'altro, permette di implementare una cosa altrimenti *impossibile,* come appunto il Management Engine.

Si, proprio MINIX, il sistema operativo didattico unix-like, realizzato nel 1987 dal mitico Andrew S. Tanenbaum per insegnare come si costruiva un vero sistema operativo. Un giocattolo per insegnare, insomma.

Eppure Linus Torvalds, dopo averlo studiato, nel 1991 pubblicò un kernel nuovo, migliore e più modificabile, e chiese via Internet la collaborazione di tutti gli interessati. Sappiamo tutti come è andata (gloriosamente) avanti questa iniziativa.

Ma in un certo senso anche MINIX non si è fermato, e pur restando sostanzialmente uguale a sé stesso, si è infilato nella maggioranza delle CPU che usiamo oggi. E sta lì a far cose decise da Intel, come ad esempio il *Management Engine*, ma anche chissà cos'altro.

Si può tranquillamente dire che MINIX è il sistema operativo più installato al mondo in applicazioni commerciali. Perciò, in proporzione, Tanenbaum dovrebbe essere più ricco di di tutti i paperoni dell'informatica moderna sommati insieme.

Ovviamente, come tutti i sistemi operativi, MINIX possiede un suo filesystem, i driver per USB ed altre periferiche, uno stack TCP/IP e persino un web server. Ed ovviamente anche tutti i bachi ed i problemi di sicurezza che può avere un software nato nel 1987, ormai congelato da decenni e che comunque, una volta "*scritto*" nella CPU, non viene mai aggiornato.

Eppure questo aggeggio, ormai antidiluviano, è alla base del "*vero funzionamento*" della maggior parte delle CPU attive nei computer di questo pianeta.

Cosa mai potrebbe andare storto?

Torniamo ora in modalità "*profetessa*".

Abbiamo raccontato solo una delle caratteristiche peculiari di una particolare pezzo di silicio di Intel, ma quante altre ne esistono nello stesso chip?

Potremmo ad esempio accennare alla possibilità di aggiornare i microcodici di una CPU Intel, modificando almeno in parte ciò che questa può fare anche al livello di Ring 0. Ed i microcodici si possono infatti aggiornare, anche su una CPU in uso, anche dagli utenti, se i firmware sono firmati con le opportune chiavi crittografiche.

E quante altre caratteristiche simili esistono in tutte le altre CPU di architetture e produttori diversi, che hanno seguito altre strade, sostanzialmente parallele?

Esagerata ed inutile complessità, ben nascosta permanentemente nel silicio, in attesa di morderci, causando problemi del tutto imprevedibili, anche perché mai analizzati.

Ora, cosa potrebbe succedere se queste caratteristiche venissero utilizzate per costruire un malware, in grado di utilizzarle, ad esempio, per bloccare a comando ed in maniera irreversibile tutte le CPU del pianeta? O magari, *riscrivendone* il funzionamento in modo che facciano altre cose, magari continuando apparentemente a lavorare come prima?

Sono cose già viste e riviste; basta un furto di chiavi crittografiche, di credenziali, di documentazione riservata, tutte cose che sono all'ordine del giorno. Non serve niente di più complesso per sovvertire questi meccanismi, ed impiegarli per altri fini.

Sarebbe concettualmente possibile creare una APT, una *minaccia persistente*, un malware non rimovibile, programmato direttamente nel silicio, pronto a scattare al momento opportuno.

Sarebbe possibile creare una Cyber-arma al confronto della quale Stuxnet farebbe la figura della versione demo di *Frogger* scritta in BASIC.

Ora capite perché non solo Biden, ma anche tutti i capi di stato delle superpotenze e delle potenze più piccole, dicono che vogliono ricominciare a realizzare i chip *a casa propria*?

Quello che non viene detto, ma che è semplicemente logico, è che tutte le perversioni ormai congelate nel silicio di architetture sempre più inutilmente complesse, proprio come MINIX, **vengono già adesso certamente utilizzate da qualche parte per scrivere malware, destinato ad essere usato come cyber-arma,** con potenza di cyber-distruzione difficilmente calcolabile.

Sono le *bombe atomiche digitali* che verranno usate in qualche prossima guerra, quando una delle parti deciderà di usare davvero le armi digitali da tempo gelosamente custodite nei cyber-arsenali. Guerra che potrebbe anche essere scatenata non da uno stato-nazione, ma da un'azienda, da una organizzazione criminale o terrorista.

Le *timide* azioni passate di cyber-guerra, tutte circoscritte o "*di prova*", hanno avuto conseguenze molto limitate nel tempo e nello spazio.

Dallo sgancio di Stuxnet sull'Iran fino al blocco dell'internet satellitare in Ucraina, **quello che è finora successo sui campi di battaglia digitali del passato non è nemmeno l'ombra di quello che succederà la prima volta che una Cyber-guerra verrà scatenata sul serio.**

Stateve accuorti.

Scrivere a Cassandra—Twitter—Mastodon
Videorubrica "Quattro chiacchiere con Cassandra"
Lo Slog (Static Blog) di Cassandra
L'archivio di Cassandra: scuola, formazione e pensiero

By Marco A. L. Calamari on March 11, 2024.

Canonical link

Exported from Medium on August 27, 2025.